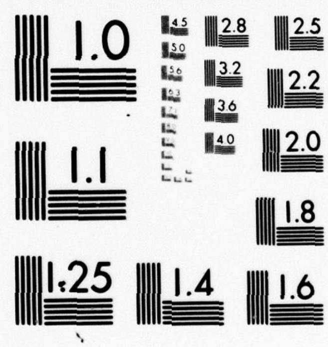
AD-A071 947 COLORADO STATE UNIV FORT COLLINS DEPT OF MATHEMATICS F/G 12/1
A COMBINED REMES - DIFFERENTIAL CORRECTION ALGORITHM FOR RATION--ETC(U)
MAR 79 E H KAUFMAN, D J LEEMING, G D TAYLOR AFOSR-76-2878

UNCLASSIFIED AFOSR-TR-79-0857 NL

| OF |

ADA
071947

END
DATE
FILMED
8-79
DDC



MICROCOPY RESOLUTION TEST CHART

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR-79-0857	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A COMBINED REMES - DIFFERENTIAL CORRECTION ALGORITHM FOR RATIONAL APPROXIMATION: EXPERIMENTAL RESULTS		5. TYPE OF REPORT & PERIOD COVERED Interim rept.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) E. H. Kaufman, Jr., D.J. Leeming ~~and~~ G. D. Taylor		8. CONTRACT OR GRANT NUMBER(s) AFOSR-76-2878, NSF-MCS78-05847
9. PERFORMING ORGANIZATION NAME AND ADDRESS Colorado State University Department of Mathematics Fort Collins, CO 80523		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304 A3
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research/NM Bolling AFB, D.C. 20332		12. REPORT DATE March 1979
		13. NUMBER OF PAGES 14
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) 16p.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Rational approximation, computation of best uniform rational approximations.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A numerical study is done comparing three algorithms for computing best rational approximations using the uniform (Chebyshev) norm. The experimental results and theoretical considerations indicate that the Remes-difcor algorithm is superior as a general-purpose routine to both the widely-used Remes algorithm and the differential correction algorithm. The three algorithms are briefly described and discussed, and the experimental results for 70 examples are presented in six tables.

3

A COMBINED REMES-DIFFERENTIAL CORRECTION ALGORITHM FOR RATIONAL APPROXIMATION: EXPERIMENTAL RESULTS

by

E. H. Kaufman, Jr., D. J. Leeming and G. D. Taylor

AMS(MOS) subject classification (1970). Primary 41A20, 41A50, 65D15.

Send proofs to: Prof. D. J. Leeming, Department of Mathematics, University of Victoria, Victoria, B.C., Canada V8W 2Y2

ABSTRACT

A numerical study is done comparing three algorithms for computing best rational approximations using the uniform (Chebyshev) norm. The experimental results and theoretical considerations indicate that the Remes-difcor algorithm is superior as a general-purpose routine to both the widely-used Remes algorithm and the differential correction algorithm. The three algorithms are briefly described and discussed, and the experimental results for 70 examples are presented in six tables.

1. Introduction

We consider the problem of approximating a given real-valued function $f(x)$ on a finite subset T of an interval [a, b] by a rational function

$$R_n^m(x) = P_m(x)/Q_n(x) = [\sum_{j=0}^{m} a_j x^j]/[\sum_{j=0}^{n} b_j x^j] \tag{1}$$

where the integers m and n are given, $(P_m, Q_n) = 1$, $Q_n(x) > 0$ on T, and $Q_n(x)$ is normalized by taking $\max_{0 \le j \le n} |b_j| = 1$. It is desired to choose $R_n^m(x)$ to minimize the error norm $\|f - R_n^m\| \equiv \max\{|f(x) - R_n^m(x)|: x \in T\}$. We note that if $f \in C[a, b]$ and the best approximation to f on [a, b] is nondegenerate (i.e. $\min(m - \partial P_m, n - \partial Q_n) = 0$, where ∂P = degree of P), then the best approximation on T can be made arbitrarily close everywhere on [a, b] to the best approximation on [a, b] by taking T sufficiently dense in [a, b] ([3]). Thus in many cases the continuous problem can be effectively treated by discretization to a finite subset.

Research supported in part by National Research Council of Canada under grant A8061, by a University of Victoria Faculty Research grant, by the Air Force Office of Scientific Research, Air Force Systems Command, USAF,.under grant AFOSR-76-2878C, and by the National Science Foundation under grant MCS-78-05847.

If a computer installation wished to have a single library routine for calculating best uniform rational approximations, then according to Lee and Roberts [7] they should select either the Remes or differential correction algorithm. To be safe they should select the differential correction algorithm because of its guaranteed convergence properties; however, the costs of this choice are a significantly slower routine than the Remes algorithm and increased storage requirements. The main purpose of this paper is to provide evidence that the Remes-difcor algorithm ([4], [8]) would be a better choice than either Remes or differential correction as a general purpose algorithm.

The three algorithms and convergence theorems are discussed in Section 2 of this paper. The numerical examples are given in Section 3, and conclusions are drawn in Section 4.

2. The Algorithms

The version of the Remes algorithm considered here is the same as considered by Lee and Roberts ([7]). This algorithm is based upon alternation. At the completion of the $(k-1)^{st}$ iteration it produces a "reference set" $X_k = \{x_1^k, x_2^k, \ldots, x_{m+n+2}^k\} \subseteq T$. In the k^{th} iteration, the algorithm first calculates (if possible) the best approximation P_k/Q_k on X_k using Newton's method to solve the nonlinear system of equations for unknowns $a_0, \ldots, a_m, b_1, \ldots, b_n$ (the coefficients of P and Q) and λ

$$f(x_i^k) - P(x_i^k)/Q(x_i^k) = (-1)^i\lambda, \quad i = 1, \ldots, m+n+2, \text{ with } b_0 = 1 \qquad (2)$$

or, equivalently,

$$f(x_i^k)Q(x_i^k) - P(x_i^k) = (-1)^i\lambda Q(x_i^k), \quad i = 1, \ldots, m+n+2, \text{ with } b_0 = 1. \qquad (3)$$

If P_k/Q_k is not the best approximation on T, the next reference set, X_{k+1}, is constructed from the extreme points of $f - P_k/Q_k$. This is done by selecting $x_1^{k+1} < \ldots < x_{m+n+2}^{k+1}$ each in T, such that $\mathrm{sgn}(f(x_i^{k+1}) - P_k(x_i^{k+1})/Q_k(x_i^{k+1}))$ $= -\mathrm{sgn}(f(x_{i+1}^{k+1}) - P_k(x_{i+1}^{k+1})/Q_k(x_{i+1}^{k+1}))$, $i = 1, \ldots, m + n + 1$, $|f(x_i^{k+1}) - P_k(x_i^{k+1})/Q_k(x_i^{k+1})| \geq \max_i |f(x_i^k) - P_k(x_i^k)/Q_k(x_i^k)|$, $i = 1, \ldots,$ $m + n + 2$, and for some η, $1 \leq \eta \leq m + n + 2$, $|f(x_\eta^{k+1}) - P_k(x_\eta^{k+1})/Q_k(x_\eta^{k+1})|$ $= \|f - P_k/Q_k\|$. This is normally referred to as a multiple exchange. The initial reference set is chosen to be the points of T which are closest to the extreme points of the $(m + n + 1)^{st}$ Chebyshev polynomial translated to [a, b]. This algorithm can fail to converge; this failure can be caused by the desired solution having a reference set of less than $m + n + 2$ points, by failure to be able to solve the nonlinear system for acceptable P_k/Q_k, or by making an improper exchange due to the existence of poles of P_k/Q_k off of the reference set X_k. It has been shown by Burke ([3]), however, that if the best approximation to f on [a, b] is nondegenerate, T is sufficiently dense in [a, b], and the choice of initial reference set is sufficiently good, then the algorithm will converge to the best approximation on T.

The differential correction program used in this paper is an improved version of a code which appeared in [6]. Recently, another version of [6] has appeared [2] which is claimed to be about 30% faster and a little more robust than the code in [6]. In order to describe the differential correction algorithm, let us assume that at the $(k-1)^{st}$ step an approximation P_k/Q_k has been calculated with $\Delta_k = \|f - P_k/Q_k\|$. The algorithm then proceeds to calculate the next approximation, P_{k+1}/Q_{k+1}, by using linear programming to solve the problem

$$\begin{cases} \text{minimize} \max\limits_{x \in T} \dfrac{|f(x)Q(x) - P(x)| - \Delta_k Q(x)}{Q_k(x)} \\ \text{subject to } |b_j| \leq 1,\ j = 0, 1, \ldots, n. \end{cases} \tag{4}$$

The initial approximation is found by minimizing $\max\limits_{x \in T} |fQ - P|$ with $b_0 = 1$, with extra constraints added to ensure that $Q > 0$ on T. It has been shown by Barrodale, Powell, and Roberts ([1]) that the algorithm has guaranteed monotone convergence; that is, $\Delta_k \downarrow \inf \|f - R_n^m\|$ even if no best approximation on T exists.

The Remes-difcor algorithm ([4], [8]) is a hybrid of the two algorithms described above. It differs from the Remes algorithm described above in two crucial respects. First, approximations on reference sets are found using the differential correction algorithm rather than by solving a nonlinear system of equations. Thus, an approximation with a positive denominator on the reference set is guaranteed even if the system has no solution. Second, if a g-pole (that is, a point where the denominator is very small in absolute value or negative) occurs somewhere in T off the reference set, and $f - P_k/Q_k$ changes sign $m + n + 1$ times on the reference set, then the next reference set is expanded to include the point where Q_k is smallest. Note that the flexibility achieved by using differential correction on the reference set is essential here, since the new reference set will have $m + n + 3$ points instead of $m + n + 2$. One point will be deleted from this enlarged reference set after P_{k+1}/Q_{k+1} is computed. The Remes-difcor algorithm will converge regardless of the choice of initial reference set, providing that g-pole free best approximations exist on all reference sets encountered.

3. Numerical Examples

The three algorithms have been run on each of fourteen functions with given sets T (see Table I) using rational functions of the form P_0/Q_2, P_1/Q_1,

P_2/Q_2, P_1/Q_3, and P_4/Q_2. The first eleven functions are the functions used by Lee and Roberts [7]. The next two functions were designed to give degenerate approximations. (Note that $\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ is a set of alternating extreme points for $f_{12}(x)-1/(1+x)$ and $\{0, .1, .2, \ldots, .9, 1\}$ is a set of alternating extreme points for $f_{13}(x) - 1/[(11/20) - x]^2$.) The last function provides examples where best approximations do not exist.

All computations were done with the University of Victoria IBM 370/145 computer using double precision arithmetic (15 digits). The Remes algorithm was terminated when the last two reference sets agreed. The differential correction algorithm was terminated when $(\|f - P_k/Q_k\| - \|f - P_{k+1}/Q_{k+1}\|)/\|f - P_k/Q_k\|$ was $\leq 10^{-7}$, or after 50 iterations. The Remes-difcor algorithm was terminated when there were no g-poles and the maximum absolute error on the reference set was within 10^{-10} of the maximum absolute error on the reference set, or after 20 exchanges.

Table II contains a summary of the results. In Tables IIIa, b, and c we give the CPU times (in seconds), and point out any examples where the computed approximation had a pole in [a, b]. In Table IV we give the error norms computed with the differential correction algorithm; the other algorithms produced the same error norms and best approximations whenever they produced pole-free approximations.

4. Conclusions

From Table II we observe that Remes-difcor was nearly as robust as differential correction and much more robust than Remes on the set of examples run. The one failure of Remes-difcor was due to cycling of best approximations, which occurred because of the apparent nonexistence of a best approximation on some of the reference sets.

Remes-difcor, although not as fast as Remes, was considerably faster than differential correction. Since the size of the point set over which approximations are actually computed at each stage is independent of the size of T for Remes and Remes-difcor, it is not surprising that as the size of T increased the ratio of differential correction time to Remes-difcor time increased, while the ratio of Remes-difcor time to Remes time decreased.

Differential correction does have the advantage of flexibility; since an alternating theory is not required, it can be used for such things as simultaneous approximation (see [5]) and approximation of functions of several variables. This provides an additional argument for Remes-difcor versus Remes, since a Remes-difcor program contains the differential correction subroutines, which can be called directly for problems on which Remes-difcor is inapplicable or fails. On the other hand, Remes-difcor can handle many problems for which differential correction cannot be used (because of storage problems) or would take excessive computer time.

In summary, Remes-difcor combines maximum robustness with good speed, and appears to us to be the best general purpose uniform rational approximation algorithm available today. A FORTRAN listing is available from the first author.

Acknowledgements

The authors would like to thank Mr. K. Bruce Wilson and Mr. Martin Milner for their valuable programming assistance.

TABLE I

Data Sets Used in the Numerical Study

Function f(x)	$[x_1, x_N]$	Number of points (Equally spaced)
f_1: e^x	$[-1,1]$	51
f_2: $\sin(x)$	$[-3,3]$	21
f_3: $\sqrt{x}$	$[0,1]$	11
f_4: $\begin{cases} 1 \\ 0 \\ -1 \end{cases}$	$[0,0.5)$ $x = 0.5$ $(0.5,1]$	21
f_5: $\begin{cases} x \\ 0.5x+0.4 \end{cases}$	$[0,1]$ $(1,2]$	51
f_6: $\begin{cases} e^x \\ e^{-x}-e^{-1}+e \end{cases}$	$[0,1]$ $(1,2]$	21
f_7: $\log(1+x)$	$[0,1]$	51
f_8: $\text{erf}(x)$	$[0,2]$	21
f_9: e^{-x^2}	$[0,2]$	11
f_{10}: $\Gamma(x)$	$[2,3]$	51
f_{11}: $\Gamma(x)$	$[2,3]$	101
f_{12}: $\begin{cases} -24x/5+3/2 \\ 52x/15-17/30 \\ -92x/21+47/14 \\ 26x/7-19/7 \end{cases}$	$[0,\frac{1}{4}]$ $(\frac{1}{4},\frac{1}{2}]$ $(\frac{1}{2},\frac{3}{4}]$ $(\frac{3}{4},1]$	21
f_{13}: $\begin{cases} (11/20-k/10)^{-2} + 0.1(-1)^k \\ \text{straight line joining adjacent points} \end{cases}$	$x = k/10$ $(k = 0,1,\ldots,10)$ $x \in [0,1],\ x \neq k/10$	11
f_{14}: $\begin{cases} 0 \\ 1 \end{cases}$	$[0,1)$ $x = 1$	21

TABLE II

Summary of Results

Algorithm	No.of failures (total = 70)	No.best approx. w/ poles in $[x_1, x_N]$	Avg.CPU time (sec)†	Avg CPU time(sec)††
Remes	13	7	0.66	.93
Differential Correction	0	4	4.40	7.82
Remes-difcor	1	4	2.38	2.31

†average based on the 57 examples in which all three algorithms converged.

††average based on the 23 examples in which all three algorithms converged and T contained 51 or more points.

Symbols used in Tables IIIa, IIIb, IIIc

F = algorithm fails to converge.

* = algorithm converges to an approximation with a pole in [a, b].

TABLE II-I a

The Remes Algorithm

CPU Times (in seconds)

	P_1/Q_1	P_0/Q_2	P_2'/Q_2	P_1/Q_3	P_4/Q_2
f_1	0.75	0.72	0.83	0.82	0.65
f_2	0.24	F	0.43*	0.50*	0.30
f_3	0.27	0.17	0.41	0.39	0.60
f_4	0.20	F	0.56	0.53	0.92
f_5	F	0.98	F	1.21	2.36
f_6	0.27	0.40	0.70	0.58	1.24
f_7	0.47	0.72	0.59	0.56	1.04
f_8	0.24	0.24	0.50	0.46	0.45
f_9	0.26	0.20	0.36	0.39	0.33
f_{10}	0.73	0.69	0.56	0.56	0.76
f_{11}	1.38	1.36	1.01	1.56	1.10
f_{12}	0.58*	F	F	0.60*	F
f_{13}	0.29	0.38*	1.16*	F	0.99*
f_{14}	F	F	F	F	F

TABLE IIIb

The Differential Correction Algorithm

CPU times (in seconds)

	P_1/Q_1	P_0/Q_2	P_2/Q_2	P_1/Q_3	P_4/Q_2
f_1	1.36	0.90	2.37	4.13	5.11
f_2	0.64	0.60	3.13	4.79	2.58
f_3	0.54	1.15	1.08	1.31	2.12
f_4	1.30	1.31	5.00	2.66	3.57
f_5	3.00	3.50	7.75	5.51	19.77
f_6	1.38	0.97	2.50	4.43	5.41
f_7	2.28	5.93	5.34	5.46	11.05
f_8	1.14	1.44	2.26	2.35	3.56
f_9	0.56	0.87	1.47	1.40	1.63
f_{10}	2.47	2.65	5.54	12.70	12.98
f_{11}	5.03	5.09	10.51	23.87	26.36
f_{12}	1.99	1.24	3.12	2.81	7.67
f_{13}	0.82	0.78*	1.64*	2.91*	1.52*
f_{14}	2.15	2.14	3.52	3.69	6.00

TABLE IIIc

The Remes-difcor Algorithm

CPU times (in seconds)

	P_1/Q_1	P_0/Q_2	P_2/Q_2	P_1/Q_3	P_4/Q_2
f_1	0.86	0.97	1.82	1.68	2.43
f_2	0.33	0.10	2.87	6.42	1.01
f_3	0.72	1.03	1.93	2.01	4.51
f_4	0.64	0.18	4.10	2.53	5.25
f_5	1.06	2.06	4.66	3.19	13.31
f_6	0.59	0.81	2.70	3.95	7.14
f_7	0.46	1.84	1.13	1.14	3.39
f_8	0.60	0.87	2.24	2.10	2.49
f_9	0.79	0.82	2.33	2.32	2.95
f_{10}	0.71	0.90	1.21	3.30	2.63
f_{11}	0.78	0.90	1.30	4.82	2.26
f_{12}	2.22	1.21	2.76	2.97	F
f_{13}	0.68	0.90*	2.59*	3.48*	7.19*
f_{14}	1.08	1.38	4.01	2.59	4.95

TABLE IV

Errors of Best Approximation

	P_1/Q_1	P_0/Q_2	P_2/Q_2	P_1/Q_3	P_4/Q_2
f_1	0.20932(-1)	0.34791(-1)	0.86644(-4)	0.12392(-3)	0.21037(-6)
f_2	0.62542(0)	0.99749(0)	0.30608(0)	0.30608(0)	0.66482(-2)
f_3	0.36243(-1)	0.18078(0)	0.77019(-3)	0.37281(-2)	0.10202(-4)
f_4	0.81818(0)	0.10000(1)	0.26923(0)	0.26923(0)	0.70465(-1)
f_5	0.58916(-1)	0.22594(0)	0.54260(-1)	0.48581(-1)	0.18717(-1)
f_6	0.30872(0)	0.20697(0)	0.86504(-1)	0.95354(-1)	0.30919(-1)
f_7	0.85978(-3)	0.92869(-1)	0.17028(-5)	0.74224(-5)	0.58255(-8)
f_8	0.44085(-1)	0.19844(0)	0.13754(-2)	0.92931(-3)	0.44515(-4)
f_9	0.72164(-1)	0.69042(-1)	0.25586(-2)	0.41421(-2)	0.38213(-4)
f_{10}	0.64376(-2)	0.64307(-2)	0.36395(-4)	0.55096(-4)	0.17428(-6)
f_{11}	0.64420(-2)	0.64351(-2)	0.36432(-4)	0.55160(-4)	0.17660(-6)
f_{12}	0.50000(0)	0.50000(0)	0.50000(0)	0.50000(0)	0.10186(0)
f_{13}	0.19754(+3)	0.10000(0)	0.10000(0)	0.10000(0)	0.10000(0)
f_{14}	0.28934(-7)	0.31263(-7)	0.41851(-7)	0.16880(-7)	0.84017(-7)

REFERENCES

1. I. Barrodale, M. J. D. Powell, and F. D. K. Roberts, The differential correction algorithm for rational ℓ_∞ approximation, SIAM J. Numer. Anal. 9 (1972), 493-504.

2. I. Barrodale, F. D. K. Roberts and K. B. Wilson, An efficient computer implementation of the Differential Correction algorithm for rational approximation, University of Victoria, Department of Mathematics Report DM-115-IR, January 1978. (Presented at the Manitoba Conference on Numerical Mathematics and Computing, Sept. 1977.)

3. M. E. Burke, Nonlinear best approximation on discrete sets, J. Approx. Theory 16 (1976), 133-141.

4. E. H. Kaufman, Jr., D. J. Leeming and G. D. Taylor, A combined Remes-Differential Correction Algorithm for Rational Approximation, Math. Comp. 32 (1978), 233-242.

5. E. H. Kaufman, Jr. and G. D. Taylor, An application of linear programming to rational approximation, Rocky Mtn. J. of Math. 4 (1974), 371-373.

6. E. H. Kaufman, Jr. and G. D. Taylor, Uniform rational approximation of functions of several variables, Internat. J. Numer. Methods Engrg. 9 (1975), 297-323.

7. C. M. Lee and F. D. K. Roberts, A Comparison of algorithms for rational ℓ_∞ approximation, Math. Comp. 27 (1973), 111-121.

8. D. J. Leeming and G. D. Taylor, Approximation with reciprocals of polynomials on compact sets, J. Approx. Theory 21 (1977), 269-280.

Department of Mathematics
Central Michigan University
Mount Pleasant, MI 48859

Department of Mathematics
University of Victoria
Victoria, B.C., Canada V8W 2Y2

Department of Mathematics
Colorado State University
Fort Collins, CO 80523